

**REMARKS**

Applicant respectfully requests reconsideration and allowance of the subject application. Claims 1-30 and 34-37 are pending in this application.

**35 U.S.C. § 102**

Claims 1-9, 20-30, and 34-37 stand rejected under 35 U.S.C. §102(b) as being unpatentable over U.S. Patent No. 5,410,646 to Tondevold et al. (hereinafter "Tondevold"). Applicant respectfully submits that claims 1-9, 20-30, and 34-37 are not anticipated by Tondevold.

Tondevold is directed to a system and method for creating, processing, and storing forms electronically (see, col. 1, lines 10-17). In Tondevold, a computer comprises a central processing unit, a display device, an input device, and an addressable memory (see, col. 3, lines 38-40). The processing unit displays images of blank forms on the display device, the blank forms including several fields and their respective labels (see, col. 4, lines 8-12). A database of form definitions is used as the source for the information and the format for the image displayed (see, col. 4, lines 29-31). The form definition specifies the form type, the number of fields, the label for each field, the data type for each field, the protection level for each field, and any interdependencies between the fields (see, col. 4, lines 31-34). The system also uses protection levels to modify the display depending on the user's identity and the protection level for the field (see, col. 5, lines 1-4). For example, a field may be designated in the form definition to be hidden from view for the district manager while displayed and modifiable for the area manager (see, col. 5, lines 9-12).

In contrast, amended claim 1 recites:

A method implemented in a computer, the method comprising:  
receiving an indication of a desired form to be used for data input;  
automatically identifying one or more data input fields to be included on the form; and  
generating, after automatically identifying the one or more data input fields, a form definition including the automatically identified one or more data input fields.

Applicant respectfully submits that no such receiving, identifying, and generating is disclosed in Tondevold.

As discussed above, Tondevold discloses a database of form definitions used as the source for the information and format for the form image displayed, and further discloses that protection levels are used to modify the display. However, this modification of the display of Tondevold is a modification of the form image displayed (different fields can be displayed as the form image based on the user's identity and protection level for the field), not a modification of the form definitions in the database of Tondevold. No discussion or mention is made in Tondevold of generating a form definition including the automatically identified one or more data input fields **after** automatically identifying the one or more data input fields. As the protection levels of Tondevold are used to modify the display, any form definition of Tondevold would already be generated prior to the modification of the display, and thus could not be generated after automatically identifying one or more data input fields. Accordingly, Applicant respectfully submits that Tondevold cannot disclose generating, after automatically identifying the one or more data input fields, a form definition including the automatically identified one or more data input fields as recited in amended claim 1.

For at least these reasons, Applicant respectfully submits that amended claim 1 is allowable over Tondevoid.

Given that claims 2-9 depend from amended claim 1, Applicant respectfully submits that claims 2-9 are likewise allowable over Tondevoid for at least the reasons discussed above with respect to amended claim 1.

With respect to amended claim 20, Applicant respectfully submits that, similar to the discussion above regarding amended claim 1, Tondevoid does not disclose determining one or more attributes that are used by a business logic but not obtained by the business logic elsewhere and using, after determining the one or more attributes, each of the one or more attributes to define a field of a form definition, the field being used to obtain data input as recited in amended claim 20.

For at least these reasons, Applicant respectfully submits that amended claim 20 is allowable over Tondevoid.

Given that claims 21-25 depend from amended claim 20, Applicant respectfully submits that claims 21-25 are likewise allowable over Tondevoid for at least the reasons discussed above with respect to amended claim 20.

With respect to amended claim 26, amended claim 26 recites:

A system comprising:

a tag library to store validation code that, when included in a form definition and executed from the form definition, verifies that an input to an associated data input field of the form defined by the form definition satisfies one or more restrictions; and

a form processor configured to automatically identify one or more restrictions to be associated with a data input field of the form, and further configured to add to the form definition, after the automatic identification of the one or more restrictions, validation code from the tag library to verify that a subsequent input to the data field satisfies the one or more automatically identified restrictions.

Applicant respectfully submits that Tondevold does not disclose a form processor as recited in amended claim 26.

Tondevold discloses that the processing unit compares the data input to the data type such as numerical, character, calculation or decimal in the form definition for a match before information will be accepted (see, col. 4, lines 61-64). Tondevold also discloses that a field may specify that the input be within a valid range or from within a data base (see, col. 4, lines 64-66), and that the processing unit verifies the correctness of data entered by the user by comparing it to a valid range of values for a particular field (see, col. 7, lines 30-35).

In contrast to Tondevold, the form processor of amended claim 26 is configured to automatically identify one or more restrictions to be associated with a data input field of the form, and further configured to add to the form definition validation code to verify that a subsequent input to the data field satisfies the one or more automatically identified restrictions. Applicant respectfully submits that there is no disclosure in Tondevold of a form processor configured to perform both the automatic identification and the verification as recited in amended claim 26.

As discussed above, Tondevold discloses a database of form definitions used as the source for the information and format for the form image displayed, and further discloses that protection levels are used to modify the display. However, this modification of the display of Tondevold is a modification of the form image displayed (different fields can be displayed as the form image based on the user's identity and protection level for the field), not a modification of the form definitions in the database of Tondevold. Any modification in Tondevold is

of the form image displayed, not the form definitions, so Tondevold cannot disclose a form processor to include validation code in a form definition. Nowhere in Tondevold is there any discussion or mention of a form processor to automatically identify one or more restrictions and to include validation code in the form definition to verify that a subsequent input to the data field satisfies the automatically identified one or more restrictions. As there is no such discussion or mention in Tondevold, Applicant respectfully submits that Tondevold cannot disclose a form processor as recited in amended claim 26.

Furthermore, Applicant respectfully submits that no discussion or mention is made in Tondevold of adding to a form definition, after one or more restrictions to be associated with a data input field of the form are automatically identified, validation code from the tag library to verify that a subsequent input to the data field satisfies the one or more automatically identified restrictions. As discussed above, Tondevold discloses a database of form definitions used as the source for the information and format for the form image displayed, and further discloses that protection levels are used to modify the display. However, this modification of the display of Tondevold is a modification of the form image displayed (different fields can be displayed as the form image based on the user's identity and protection level for the field), not a modification of the form definitions in the database of Tondevold. As the protection levels of Tondevold are used to modify the display, any form definition of Tondevold would already be generated prior to the modification of the display, and thus would not have added to it validation code after automatically identifying one or more data input fields. Accordingly,

Applicant respectfully submits that Tondevold cannot disclose a form processor as recited in amended claim 26.

For at least these reasons, Applicant respectfully submits that claim 26 is allowable over Tondevold.

Given that claims 27-30 depend from claim 26, Applicant respectfully submits that claims 27-30 are likewise allowable over Tondevold for at least the reasons discussed above with respect to claim 26.

With respect to amended claim 34, amended claim 34 recites:

An architecture comprising:  
a business logic layer to process requests received from a client; and  
an execution environment layer via which a form processing module can communicate with the business logic layer, wherein the form processing module obtains, from the business logic layer, an indication of one or more restrictions on data input to a form for a request to be subsequently processed by the business logic layer, and adds the one or more restrictions to a form definition for the form.

Applicant respectfully submits that Tondevold does not disclose an architecture as recited in amended claim 34.

In amended claim 34, a form processing module communicates with a business logic layer via an execution environment layer, and obtains an indication of one or more restrictions on data input to a form from the business logic layer, and also adds the one or more restrictions to a form definition for the form. Applicant respectfully submits that there is no disclosure of such a business logic layer in Tondevold. In the July 25, 2005 Office Action at p. 10, the accepting input from a user and validating the various data type input by the user using form definitions found in memory is cited as teaching the business logic layer of claim 34. However, there is no disclosure in Tondevold of the accepting and

validating of Tondevold being the source for the one or more restrictions provided to an execution environment layer. Without any such disclosure, Applicant respectfully submits that Tondevold cannot disclose a form processing module obtains, from the business logic layer, an indication of one or more restrictions on data input to a form for a request to be subsequently processed by the business logic layer as recited in amended claim 34.

Furthermore, there is no disclosure in Tondevold of that same form processing module that obtains the indication of the one or more restrictions from the business logic layer also identifying the restrictions in a form definition for the form. In order to satisfy the language of claim 34, Tondevold would need to disclose a form processing module that **both** obtains an indication of one or more restrictions on data input to a form from the business logic layer, and adds those one or more restrictions to a form definition for the form. However, Applicant respectfully submits that there is no disclosure of a form processing module in Tondevold which performs both of these functions. As discussed above, there is no disclosure of a business logic layer being a source for the one or more restrictions, and thus Tondevold cannot disclose a form processing module that obtains an indication of one or more restrictions on data input to a form from the business logic layer. As such indication cannot be obtained in Tondevold, there cannot be any identification of such one or more restrictions in a form definition for the form in Tondevold because it would not be possible to identify the restrictions in the form definition when the restrictions cannot be obtained in Tondevold.

In addition, Applicant respectfully submits that no discussion or mention is made in Tondevold of adding to a form definition one or more restrictions on data input to a form, an indication of the one or more restrictions being obtained from a business logic layer. As discussed above, Tondevold discloses a database of form definitions used as the source for the information and format for the form image displayed, and further discloses that protection levels are used to modify the display. However, this modification of the display of Tondevold is a modification of the form image displayed (different fields can be displayed as the form image based on the user's identity and protection level for the field), not a modification of the form definitions in the database of Tondevold. As the protection levels of Tondevold are used to modify the display, any form definition of Tondevold would already be generated prior to the modification of the display, and thus would not have added to it one or more restrictions, an indication of the one or more restrictions having been obtained from a business logic layer. Accordingly, Applicant respectfully submits that Tondevold cannot disclose an execution environment layer as recited in amended claim 34.

For at least these reasons, Applicant respectfully submits that amended claim 34 is allowable over Tondevold.

Given that claim 35 depends from claim 34, Applicant respectfully submits that claim 35 is likewise allowable over Tondevold for at least the reasons discussed above with respect to claim 34.

With respect to claim 36, claim 36 recites:

A method comprising:  
accessing a business logic to identify one or more interactions  
associated with the business logic, wherein each interaction is



associated with a request and includes one or more command definitions to process the request;

identifying, in the one or more interactions, one or more attributes that are not obtained by the one or more interactions elsewhere; and

indicating that the one or more identified attributes are to be obtained via a data input field on a form, and further indicating that an input for the data input field is needed when submitting the form.

Applicant respectfully submits that Tondevold does not disclose the accessing, identifying, and indicating of claim 36.

In the July 25, 2005 Office Action at p. 10, it was asserted that the attributes that are not obtained by the one or more interactions elsewhere of claim 36 is taught by the comparing the input made to the forms with the user's identity, such as an area manger of Tondevold. Applicant respectfully disagrees and submits that identifying, in the one or more interactions, one or more attributes that are not obtained by the one or more interactions elsewhere is not disclosed by Tondevold.

Claim 36 recites indicating that the one or more identified attributes are to be obtained via a data input field on a form, and further indicating that an input for the data input field is needed when submitting the form. If the comparing the input made to the forms with the user's identity of Tondevold teaches the identifying one or more attributes of claim 36 as asserted in the July 25, 2005 Office Action, then Tondevold would also have to disclose indicating that the user's identity is to be obtained via a data input field on a form. Although Tondevold, as discussed above, discloses use of protection levels to modify the display depending on the user's identity and the protection level for the field, such as a field that may be designated in the form definition to be hidden from view for the district manager while displayed and modifiable for the area manager, there is

no discussion or mention in Tondevold of indicating that the user's identity is to be obtained via a data input field on a form. In fact, the user's identity could not be obtained via a data input field on a form of Tondevold because the fields that are displayed on the form of Tondevold are modified depending on the user's identity; the user's identity is thus necessary for display of the form, so the user's identity could not be obtained from the form. As such, Applicant respectfully submits that Tondevold cannot disclose identifying, in the one or more interactions, one or more attributes that are not obtained by the one or more interactions elsewhere as recited in claim 36.

For at least these reasons, Applicant respectfully submits that claim 36 is allowable over Tondevold.

Given that claim 37 depends from claim 36, Applicant respectfully submits that claim 37 is likewise allowable over Tondevold for at least the reasons discussed above with respect to claim 36.

Applicant respectfully requests that the §102 rejections be withdrawn.

### **35 U.S.C. § 103**

Claims 10-19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Tondevold in view of U.S. Patent No. 6,704,906 to Yankovich et al. (hereinafter "Yankovich"). Applicant respectfully submits that claims 10-19 are not obvious over Tondevold in view of Yankovich.

Yankovich is directed to a self-directed routable electronic form system and method (see, Title). As discussed in the Abstract of Yankovich, each user in the process defines the next, or all the subsequent user routing. The invention resides

in providing rules and guidelines that define routing to exist completely within the electronic form itself. Residing in the electronic form these routing rules are dynamically changed based on data input to the form, providing visual clues to the routing based on that input. Yankovich generates in the approvers' view of the form appropriate visual elements corresponding with allowable actions of this subsequent viewer. Yankovich also provides visual clues as to the next routing destination based on the electronics form current fill-in data, or based on any changes the approver makes to fill-in data, all without server interaction. This sequence continues until the electronics "form flow" is complete and an end point is reached.

With respect to amended claim 10, Applicant respectfully submits that, similar to the discussion above regarding amended claim 1, Tondevoid does not disclose or suggest automatically identifying one or more restrictions associated with a data input field and using, after automatically identifying the one or more restrictions, the one or more restrictions and the field to generate a text markup language form definition as recited in amended claim 10. Yankovich is cited in the July 25, 2005 Office Action at p. 11 as teaching the creation of a form in HTML. Yankovich is not cited as curing, and does not cure, the deficiencies of Tondevoid discussed above with respect to amended claim 1. Accordingly, Applicant respectfully submits that amended claim 10 is allowable over Tondevoid in view of Yankovich for at least these reasons.

Given that claims 11-19 depend from amended claim 10, Applicant respectfully submits that claims 11-19 are likewise allowable over Tondevoid in

view of Yankovich for at least the reasons discussed above with respect to amended claim 10.

Applicant respectfully requests that the §103 rejections be withdrawn.

**Conclusion**

Claims 1-30 and 34-37 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. Should any matter in this case remain unresolved, the undersigned attorney respectfully requests a telephone conference with the Examiner to resolve any such outstanding matter.

Respectfully Submitted,

Date: 1/24/06

By: 

Allan T. Sponseller  
Reg. No. 38,318  
(509) 324-9256